

REMARKS

Applicants have received the Office Action mailed June 4, 2009. Applicants are not amending any claims in the present response. Claims 1-15 are pending, of which claims 1, 6, and 11 are independent. Applicants request reconsideration of the pending claims in view of the following remarks.

Claim Rejections – 35 U.S.C. § 103

The Office Action (at page 2) rejected claim 1-3, 5-8, 10-13, and 15 under 35 U.S.C. § 103(a) as being unpatentable over SAP Wizard Builder in view of Kodosky et al. (U.S. Pat. No. 4,901,221). The Office Action (at page 15) rejected claims 4, 9, and 14 under 35 U.S.C. § 103(a) as being unpatentable over the SAP Wizard Builder, in view of Kodosky, in view of Watson-Luke et al (U.S. Pub. No. 2005/0114240).

These rejections are respectfully traversed. Applicants assert that even if a person of ordinary skill in the art would perform the proposed combination, it would not provide the subject matter described in the independent claims. Moreover, a skilled artisan would not have a reason to combine the SAP Wizard Builder and Kodosky, whether as asserted in the Office Action or otherwise.

Claim 1 recites:

A method to be executed as part of a process for creating an executable configuration program that comprises multiple steps that are successively executed and wherein associated with each of the steps are user-selectable options, the method comprising:

generating a user interface including at least (i) a logic flow area for a user to define a command structure for the configuration program including at least one step, (ii) a refinement area for the user to specify a configuration detail regarding a step arranged in the logic flow area, and (iii) a rule palette for the user to create a rule, wherein the rule palette provides multiple conditional operators and entry fields;

creating and storing in a repository the rule that during execution of the configuration program determines which of the user-selectable options for the step are displayed when a specified user-selectable option of a previous step is selected, wherein the step and the previous step are arranged in the logic flow area and the user specifies the rule in the refinement area; and

binding the rule in the repository to the specified user-selectable option so that during execution of the configuration program the rule is executed when the specified user-selectable option is selected. (Emphasis added to language that the Office Action relies upon Kodosky as allegedly teaching.)

The relied upon references do not teach every limitation of the independent claims

Unlike claim 1, SAP Wizard Builder and Kodosky fail to teach “***creating . . . the rule that during execution of the configuration program determines which of the user-selectable options for the step are displayed when a specified user-selectable option of a previous step is selected.***” The Office Action (at page 3) relies upon the SAP Wizard Builder as allegedly teaching the recited claim language. Specifically, the Office Action states that “rule created and binded [sic] to execute the rule binded [sic] to the specific option when user selects the user-selectable option as claimed above is shown in Figures 4-7 for example, also see annotation in figure below.” (Office Action, page 3).

Figures 4-7 of the SAP Wizard builder show a user interface for creating a wizard that includes identifying a name for the wizard (Figure 5), determining steps and creating descriptive texts for the steps (Figure 6), and assigning subscreens to the steps (Figure 7). The relied upon screenshots do not teach creating a rule that “determines which of the user-selectable options for [a] step are displayed when a specified user-selectable option of a previous step is selected.” In some implementations, the recited subject matter includes creating a rule that governs which set of user-selectable options are displayed in a screen of a wizard, the displayed user-selectable options dependent upon what user-selectable options were selected in a previous screen. (Application, at page 16, lines 7-23.) SAP Wizard builder does not disclose a system for creating such a rule. Indeed, the Office Action has failed to articulate how the relied upon screenshots teach the recited language.

Applicant notes that the Office Action references an “annotation in figure below,” however, the Office Action did not include an annotated figure (only a non-annotated screenshot of Fig. 5 in the Office Action’s discussion of claim 5). Applicant respectfully requests that the Examiner mail an another office action that fully articulates how SAP Wizard Builder allegedly teaches the recited subject matter. A claim rejection does not comply with 35 U.S.C. § 132 if it “is so uninformative that it prevents the applicant from recognizing and seeking to counter the grounds for rejection.” *Chester v. Miller*, 906 F.2d 1574, 1578 (Fed. Cir. 1990).

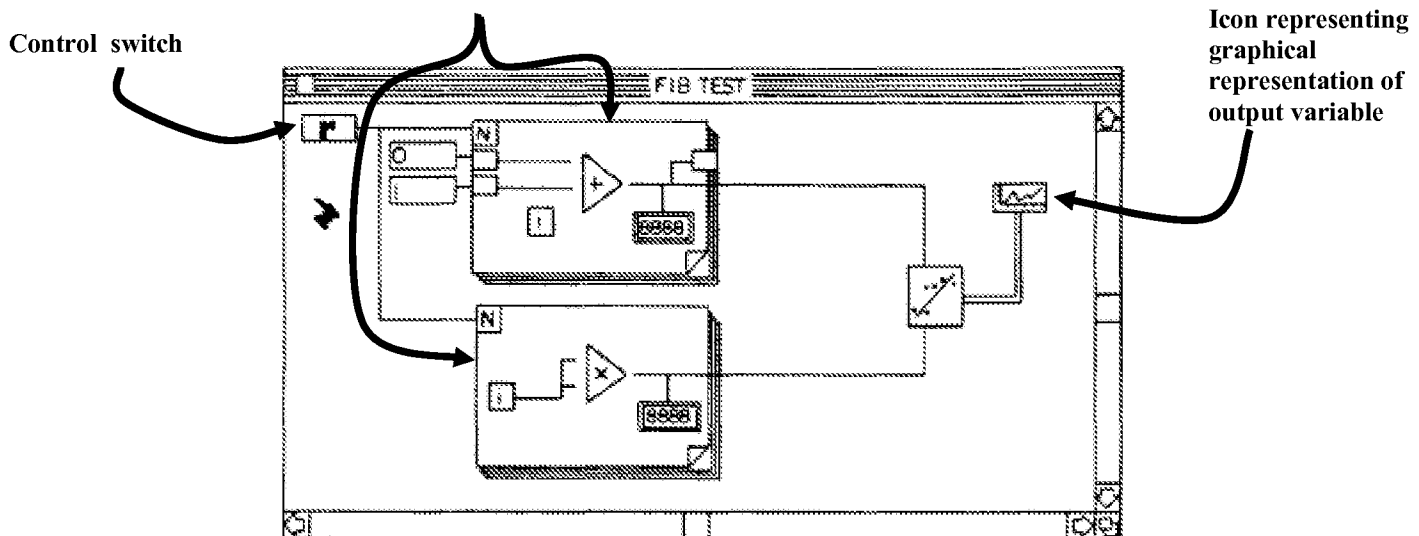
Kodosky does not provide the subject matter that SAP Wizard Builder is lacking. Kodosky describes a virtual instrument for modeling a manipulation of input variables to produce output variables, and does not teach “creating . . . the rule that during execution of the

configuration program determines which of the user-selectable options for the step are displayed when a specified user-selectable option of a previous step is selected.” (Kodosky, at col. 8, lines 3-23 and col. 4, lines 2-14.) Indeed, the Office Action did not rely upon Kodosky. Accordingly, the proposed combination fails to disclose the subject matter described in independent claim 1. Applicant respectfully submits that claim 1 is patentable over SAP Wizard Builder, Kodosky, and all other references in the record. Independent claims 6 and 11 contain similar language and are patentable for at least the same reasons. Dependent claims 2-5, 7-10, and 12-15 are patentable for at least the same reasons, and for the independently patentable features recited therein.

The Office Action has failed to set forth an “articulated reason” to combine the references

The Office Action (at page 4) concedes that the features of the recited user interface are not taught by SAP Wizard builder. The Office Action relies upon Kodosky as allegedly teaching these features (emphasized with underline in the above copy of claim 1). Kodosky describes a user interface that “permits a user to construct a virtual instrument.” (Kodosky, col. 8, lines 3-23.) A user of Kodosky’s system can virtually model a digital electrical system by wiring together “components” that represent the electrical components and adjusting “control dials and switches for providing variable input information.” (Kodosky, at col. 8, lines 3-23 and 38-64; and col. 25, line 55 to col. 26, line 45.) A graphical representation of the output variables is provided below. (*Id.*)

“Components” modeling portions of an electrical system



The Office Action states:

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify SAP and include a logic flow area, a refinement area, and a rule template area with the motivation to provide the user with a simpler method of creating the logic flow and to allow the user to more easily create a rule for execution.

The Office Action improperly relies upon conclusory statements and impermissible hindsight to purport that the combination of SAP Wizard builder and Kodosky would have been obvious. Indeed, the Kodosky reference is non-analogous art, rendering it inappropriate as a reference under 35 U.S.C. § 103.

Applicant agrees with the Office Action that the recited graphical user interface of the present application can provide a simpler method of creating a logic flow. Applicant, however, disagrees with the Office Action's contention that modifying the SAP Wizard Builder would have been obvious simply given a desire of one skilled in the art to provide a "simpler method of creating the logic flow." Indeed, other than this conclusory statement, the Office Action has failed to articulate a rationale for combining the SAP Wizard Builder with Kodosky.

"[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some *articulated reasoning* with some rational underpinning to support the legal conclusion of obviousness." *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 418 (2007). Indeed, the Office Action has impermissibly relied upon hindsight to conclude that one skilled in the art seeking to provide a simpler method of creating a logic flow would look to the teachings of the Kodosky reference. *Id.* at 421 ("A factfinder should be aware, of course, of the distortion caused by hindsight bias and must be cautious of arguments reliant upon *ex post* reasoning.").

Kodosky describes a system that modifies a user-defined input variable with "components" and graphically displays the modified output signal. In contrast, the recited subject matter relates to a system that does not model the modification of an input signal in an electrical circuit. Indeed, the recited subject matter relates to a system for creating an executable configuration program (e.g., a "wizard") for configuring software programs. (Application, at page 2, lines 11-16.) Such a system is not a "virtual instrument" and does not receive an input signal and modify the input signal to model an electrical instrument. The system can instead

define a sequence of screens that will be displayed in the wizard. Kodosky is a non-analogous reference and should not be relied upon. Indeed, the Office Action has failed to “determine[e] the scope and content of the prior art” as required by *KSR* 550 U.S., at 399 (2007). A technique is obvious if “a person of ordinary skill in the art would recognize that it would improve *similar* devices in the same way.” *KSR*, 550 U.S., at 401. In the present situation, however, the devices are not similar.

Accordingly, the proposed combination of SAP Wizard Builder and Kodosky is not a proper basis for the rejections. Applicant respectfully submits that claim 1 is patentable over SAP Wizard Builder, Kodosky, and all other references in the record. Independent claims 6 and 11 recite language that is similar to that in claim 1 and are patentable for at least the same reasons. Dependent claims 2-5, 7-10, and 12-15 are patentable for at least the same reasons as their independent claims, and for the independently patentable features therein.

Conclusions

Claims 1-15 appear to be in form for allowance. As such, Applicants request that the Examiner allow claims 1-15.

It is believed that all of the pending claims have been addressed. However, the absence of a reply to a specific rejection, issue or comment does not signify agreement with or concession of that rejection, issue or comment. In addition, because the arguments made above may not be exhaustive, there may be reasons for patentability of any or all pending claims (or other claims) that have not been expressed. Finally, nothing in this paper should be construed as an intent to concede any issue with regard to any claim, except as specifically stated in this paper, and the amendment of any claim does not necessarily signify concession of unpatentability of the claim prior to its amendment.

Please apply any charges or credits to deposit account 06-1050.

Applicant : Yuh-Cherng Wu
Serial No. : 10/829,145
Filed : April 21, 2004
Page : 12 of 12

Attorney's Docket No.: 13906-0122001 / 2003P00271 US

Respectfully submitted,

Date: August 6, 2009 _____

/j richard soderberg reg. no. 43,352/ _____
J. Richard Soderberg
Reg. No. 43,352

Fish & Richardson P.C.
3200 RBC Plaza
60 South Sixth Street
Minneapolis, Minnesota 55402
Telephone: (612) 335-5070
Facsimile: (877) 769-7945

60576499.doc